STATS 509 Homework 2 Xiaofeng Nie

1. Solution:

a)

```
> mu;sd;q
[1] 0.0006352362         > quantile(logr,0.005)
[1] 0.008922481                  0.5%
[1] -0.02841946          -0.03025071
```

From the outcome of R program, we know that the mean of log-returns is 0.00064; the standard deviation of log-returns is 0.0089; the relative value at risk is 0.0284. The simply 0.005-quantile of the log-returns is the -0.0303, so the relative value at risk is 0.0303, which is larger than the relative value at risk computed with double exponential distributed log-returns.

b) Using Monte-Carlo method, we get expected shortfall is 347504.2 dollar.

```
> mean(na.omit(exp_shortfall))*1e7
[1] -347504.2
```

c) When positive loss is single-sided exponential distributed, the relative value at risk is 0.0299 less than the relative value at risk computed with double exponential distributed log-returns.
Employing Monte-Carlo method, the expected shortfall is 365811.4 dollar.

```
                      > quantile(posloss,p_pos)
> q_pos                 98.86957%                  > mean(na.omit(exp_shortfall))*1e7
[1] 0.02991403         0.03025633                  [1] 365811.4
```

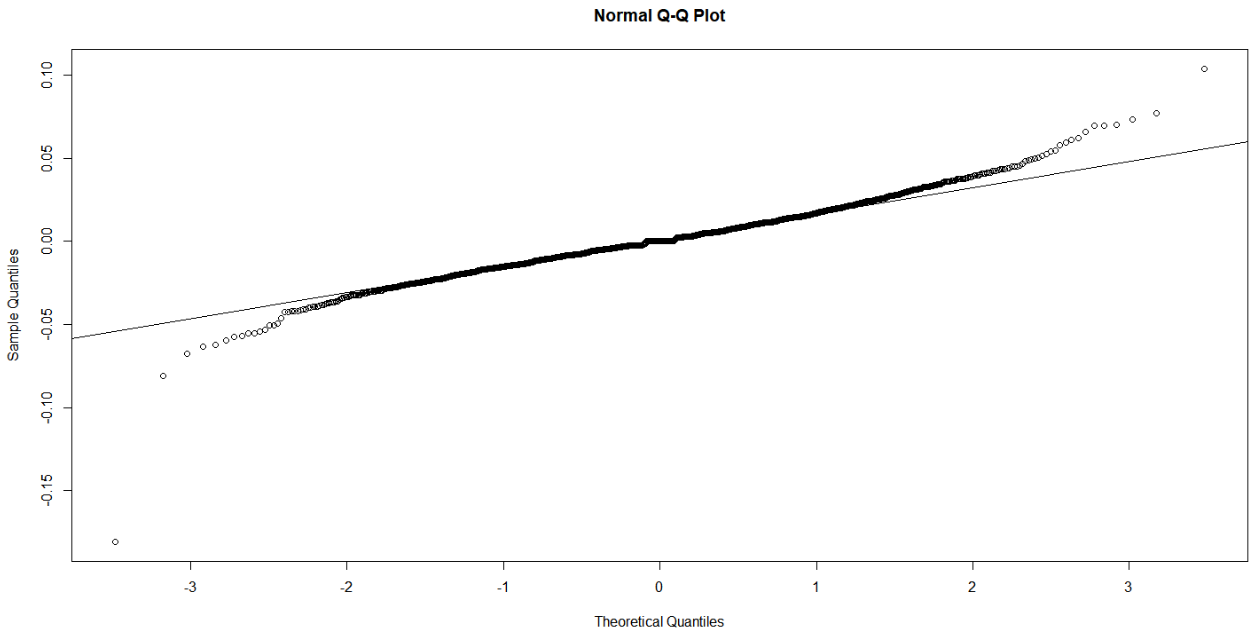
2. Solution:

a)

```
> summary(dat$FORD)
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-0.1810089 -0.0099071  0.0000000  0.0007601  0.0112899  0.1039604
> sd(dat$FORD)
[1] 0.01831557
```

The mean is -0.181; the median is 0; the standard deviation is 0.0183.

b)



Considering the normal Q-Q plot, we can say that Ford returns are not normal distributed, because they have obvious heavy tail.

c)

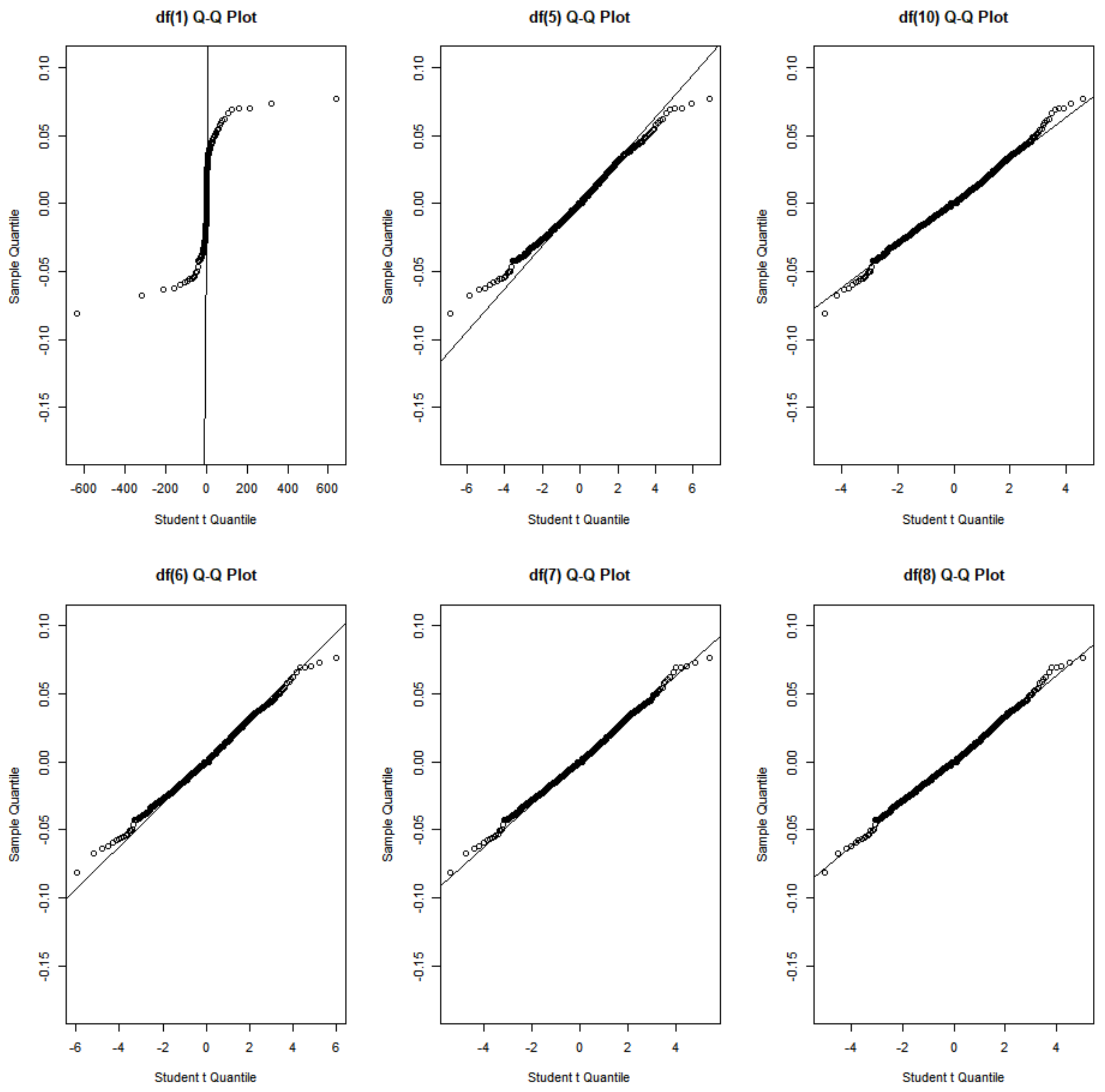
```
> shapiro.test(dat$FORD)

        Shapiro-Wilk normality test

data:  dat$FORD
W = 0.96388, p-value < 2.2e-16
```

Shapiro-Wilk test shows that the p-value is way less than 0.01, which means that we can reject the null hypothesis that these data are normal distributed, i.e. Shapiro-Wilk test thinks Ford returns are not normal distributed.

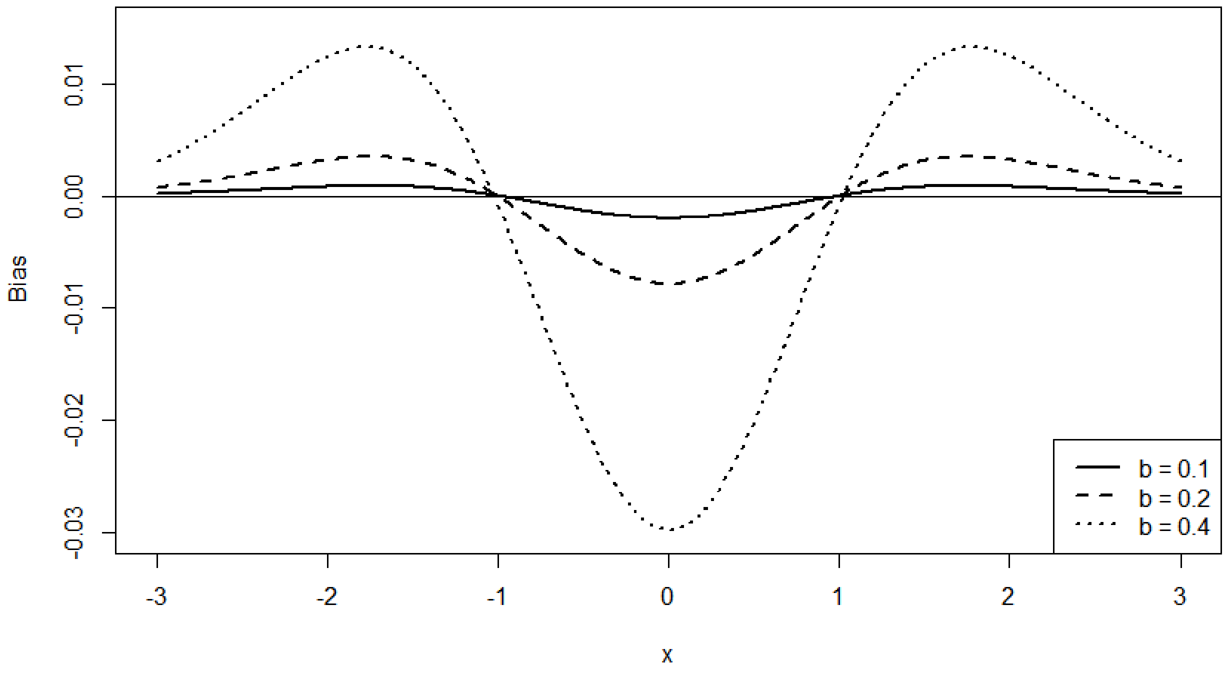
d) According to the Q-Q plot of student t distribution, we find that df=7 is the best choice.
The point of Black Monday, October 19, 1987 should be deleted. People will spend much less money than other days, so this point may be an extreme point in the data set, which will dramatically influence the distribution of the whole data. Thus, deleting this point will help us to find an optimal distribution.



e)
```
> se_mean = sd(dat$FORD)/sqrt(length(dat$FORD))
> quant = quantile(ecdf(dat$FORD),0.5)
> f = density(dat$FORD,from = quant,to = quant,n=1,kernel = "gaussian")$y
> se_median = sqrt(0.5*0.5/(length(dat$FORD)*(f)^2))
> se_mean;se_median
[1] 0.0004095486
[1] 0.0004285607
```
Using formula (4.3), we can calculate that the standard error of sample median is 0.0004285607, the standard error of sample mean is 0.0004095486. Standard error of sample median is a little larger than standard error of sample mean. But the scale of difference is 10^{-4}, which could be almost ignored, because n=2000 is large.

3. Solution:

b)



From the plot above, we can find that when bandwidth is increasing, bias will increase. When bandwidth is 0.1, bias is close to zero. When x is in the range of (-1,1), the bias is larger. And when x is large, bias converges to 0. It means that KDE is way close to the real pdf of standard normal distribution as x is large enough.

Appendix:

```
#a
qdexp <- function(p,mu,lambda){
  quant1 <- qexp(0*p,lambda) + mu
  pn <- p[p<.5]
  pp <- p[p>.5]
  quant1[p>.5] <- qexp(2*pp-1,lambda) + mu
  quant1[p<.5] <- -qexp(1-2*pn,lambda) + mu
  quant1
}
dat = read.csv("O:\\18WIN\\STATS
509\\HW1\\Nasdaq_daily_Jan1_2012_Dec31_2017.csv",header = T)
logr = diff(log(dat$Adj.Close))
mu = mean(logr)
sd = sd(logr)
lambda = sqrt(2)/sd
p = 0.005
q = qdexp(p,mu,lambda)
mu;sd;q
quantile(logr,0.005)

#b
rdexp <- function(n,mu,lambda){
  rexp <- rexp(n,lambda)
  rbin <- 2*rbinom(n,1,.5)-1
  x <- rexp*rbin+mu
}

niter = 1e4
exp_shortfall = rep(0,niter)
set.seed(2015)
for (m in 1:niter)
{
  r = rdexp(1508,mu,lambda)
  index = which(r < q)
  exp_shortfall[m] = mean(r[index])
}
mean(na.omit(exp_shortfall))*1e7

#ca
posloss = -logr[logr < 0]
n = length(posloss)
mean = mean(posloss)
std = sd(posloss)
lambda = 1 / mean
p_pos = (n - 0.005*length(logr))/n
q_pos = qexp(p_pos,rate = lambda)
q_pos
quantile(posloss,p_pos)

#cb
rdexp <- function(n,mu,lambda){
  rexp <- rexp(n,lambda)
  rbin <- 2*rbinom(n,1,.5)-1
  x <- rexp*rbin+mu
}

niter = 1e4
exp_shortfall = rep(0,niter)
set.seed(2015)
for (m in 1:niter)
{
  r = rexp(n,rate = lambda)
```

```r
  index = which(r > q_pos)
  exp_shortfall[m] = mean(r[index])
}
mean(na.omit(exp_shortfall))*1e7

dat = read.csv("O:\\18WIN\\STATS 509\\datasets\\ford.csv",header=T)
dim(ford)
summary(dat$FORD)
sd(dat$FORD)
qqnorm(dat$FORD)
qqline(dat$FORD)
shapiro.test(dat$FORD)

#d
par( mfrow = c(2, 3 ) )
q1 = qt(seq(0,1,length = 2000),df = 1)
q2 = sort(dat$FORD)
plot(q1,q2,xlab = "Student t Quantile",ylab = "Sample Quantile",main="df(1) Q-Q
Plot")
qqline(q2)
q1 = qt(seq(0,1,length = 2000),df = 5)
plot(q1,q2,xlab = "Student t Quantile",ylab = "Sample Quantile",main="df(5) Q-Q
Plot")
qqline(q2)
q1 = qt(seq(0,1,length = 2000),df = 10)
plot(q1,q2,xlab = "Student t Quantile",ylab = "Sample Quantile",main="df(10) Q-Q
Plot")
qqline(q2)
q1 = qt(seq(0,1,length = 2000),df = 6)
plot(q1,q2,xlab = "Student t Quantile",ylab = "Sample Quantile",main="df(6) Q-Q
Plot")
qqline(q2)
q1 = qt(seq(0,1,length = 2000),df = 7)
plot(q1,q2,xlab = "Student t Quantile",ylab = "Sample Quantile",main="df(7) Q-Q
Plot")
qqline(q2)
q1 = qt(seq(0,1,length = 2000),df = 8)
plot(q1,q2,xlab = "Student t Quantile",ylab = "Sample Quantile",main="df(8) Q-Q
Plot")
qqline(q2)

#e
se_mean = sd(dat$FORD)/sqrt(length(dat$FORD))
quant = quantile(ecdf(dat$FORD),0.5)
f = density(dat$FORD,from = quant,to = quant,n=1,kernel = "gaussian")$y
se_median = sqrt(0.5*0.5/(length(dat$FORD)*(f)^2))
se_mean;se_median

x = seq(-3,3,by = 0.01)
bias.1 = (pnorm(x+1.732*0.1) - pnorm(x-1.732*0.1))/(3.464*0.1) - dnorm(x)
bias.2 = (pnorm(x+1.732*0.2) - pnorm(x-1.732*0.2))/(3.464*0.2) - dnorm(x)
bias.4 = (pnorm(x+1.732*0.4) - pnorm(x-1.732*0.4))/(3.464*0.4) - dnorm(x)
plot(x,bias.1,type = "l",ylim = c(-0.03,0.015),lwd = 2,ylab = "Bias")
lines(x,bias.2,lty = 2,lwd = 2)
lines(x,bias.4,lty = 3,lwd = 2)
legend("bottomright", c("b = 0.1","b = 0.2","b = 0.4"),lty = c(1,2,3),lwd = 2)
abline(h = 0)
```